como sería el uso de esas 2 metodologias especificamente para el desarrollo de un Sistema Experto para la identificación de Algas Marinas?

Perfecto, integremos \*\*CommonKADS\*\* y \*\*Scrum\*\* para desarrollar un \*\*Sistema Experto (SE) de identificación de algas marinas\*\*. Este es un proyecto realista en biología marina, donde el conocimiento experto (taxonomía, morfología, hábitats) es complejo y requiere validación rigurosa. Aquí el plan detallado:

---

### \*\*1. Definición del Proyecto\*\*

- \*\*Objetivo\*\*: Identificar especies de algas mediante características morfológicas, color, textura, hábitat y zona geográfica.

- \*\*Inputs\*\*: Fotos, descripciones de muestras (ej: "alga parda, lobulada, en pozas de marea").

- \*\*Outputs\*\*: Especie probable + confianza (ej: "\*Fucus vesiculosus\*, 95%").

---

### \*\*2. Roles Clave Combinados\*\*

Rol y Responsabilidades

**Knowledge Owner (Experto en Algas)**: Un experto del dominio (en nuestro caso, en Algas) Valida reglas, aporta conocimiento experto, facilita claves dicotómicas, prioriza especies. Knowledge Owner: (ej: médico en un SE médico) que prioriza reglas y valida el modelo de conocimiento.

**Scrum Master:** Facilita sprints y la comunicación entre el Knowledge Owner y desarrolladores.

**Desarrolladores:** Implementan motor de inferencia, base de conocimientos e interfaz en código.

---

### \*\*3. Proceso Paso a Paso: Integrando CommonKADS + Scrum\*\*

#### \*\*Fase 0: Preliminar (CommonKADS)\*\*

- \*\*Actividad\*\*: Modelar el conocimiento inicial con experto.

- \*\*Conceptos clave\*\*:

```mermaid

classDiagram

class Alga {

+string nombre\_cientifico

+string nombre\_comun

}

class Caracteristica {

+string color

+string forma\_hoja

+string textura

+string habitat

}

Alga "1" -- "1" Caracteristica : tiene

```

- \*\*Reglas básicas\*\* (ejemplo):

```python

SI color == "pardo"

Y forma\_hoja == "vesicular"

Y habitat == "zona intermareal"

ENTONCES Alga = "Fucus vesiculosus" (confianza=0.95)

```

#### \*\*Sprint 1: Especies Prioritarias (Ej: 5 algas comunes)\*\*

| \*\*CommonKADS\*\* | \*\*Scrum\*\* |

|----------------|-----------|

| \*\*Adquisición\*\*: Entrevista con biólogo para 5 algas (ej: \*Ulva lactuca\*, \*Laminaria digitata\*).<br>\*\*Salida\*\*: 20-30 reglas validadas. | \*\*Tareas técnicas\*\*: <br>- Crear BD de algas (campos: color, forma, hábitat).<br>- Implementar motor de reglas básico (Drools/Python).<br>- Interfaz para ingresar características. |

| \*\*Validación\*\*: Probar reglas con 50 muestras de herbario digital. | \*\*Criterio de Aceptación\*\*:<br>- Sistema identifica 5 algas con >85% precisión.<br>- Tiempo respuesta < 2 seg. |

---

#### \*\*Sprint 2: Mejora de Modelo + Nuevas Especies\*\*

| \*\*CommonKADS\*\* | \*\*Scrum\*\* |

|----------------|-----------|

| \*\*Refinar conocimiento\*\*: Añadir atributos (ej: "margen\_hoja = dentado/liso").<br>\*\*Nuevas especies\*\*: Incluir 3 algas rojas (ej: \*Chondrus crispus\*). | \*\*Tareas técnicas\*\*:<br>- Ampliar BD con nuevos atributos.<br>- Añadir módulo de "confianza" en resultados.<br>- Subir fotos de referencia. |

| \*\*Validación\*\*: Pruebas con muestras ambiguas (ej: algas juveniles). | \*\*Criterio\*\*: Reducir falsos positivos a <10%. |

---

#### \*\*Sprint 3: Manejo de Incertidumbre y Excepciones\*\*

| \*\*CommonKADS\*\* | \*\*Scrum\*\* |

|----------------|-----------|

| \*\*Modelar excepciones\*\*:<br> - "SI forma\_hoja == 'linear' PERO largo > 2m ENTONCES \*Laminaria hyperborea\*".<br>\*\*Patrones difusos\*\*: Lógica para atributos subjetivos (ej: "color verde-oliva"). | \*\*Tareas técnicas\*\*:<br>- Implementar lógica difusa (bibliotecas como scikit-fuzzy).<br>- Diseñar flujo de "segunda opinión" para casos dudosos. |

| \*\*Validación\*\*: Usar 100 muestras problemáticas (ej: algas decoloradas). | \*\*Criterio\*\*: Sistema sugiere "Consulta humana" en casos ambiguos. |

---

### \*\*4. Herramientas Concretas\*\*

| Propósito | Herramientas |

|-----------|--------------|

| \*\*Modelado de Conocimiento\*\* | Protégé (ontología de algas), Tablas de decisión en Excel. |

| \*\*Gestión Ágil\*\* | Jira (con campos personalizados):<br>- `Tipo`: [Conocimiento/Técnica]<br>- `Validado por biólogo`: [Sí/No]. |

| \*\*Implementación\*\* | - Motor de reglas: \*\*Drools\*\*.<br>- Backend: Python + Flask.<br>- Base de datos: PostgreSQL (almacena reglas y muestras). |

| \*\*Validación\*\* | - Dataset público: [AlgaeBase](https://www.algaebase.org/).<br>- Muestras locales digitalizadas. |

---

### \*\*5. Ejemplo de una Regla en Acción\*\*

\*\*Caso\*\*: Usuario describe un alga:

```json

{

"color": "pardo",

"forma\_hoja": "vesicular",

"textura": "corriácea",

"habitat": "rocas intermareales"

}

```

\*\*Regla CommonKADS\*\* (en Drools):

```java

rule "Identificar Fucus vesiculosus"

when

$a : Alga(color == "pardo",

forma\_hoja == "vesicular",

textura == "corriácea",

habitat == "rocas intermareales")

then

$a.setEspecie("Fucus vesiculosus");

$a.setConfianza(0.95);

end

```

\*\*Salida del Sistema\*\*:

> "\*\*Especie identificada\*\*: \*Fucus vesiculosus\* (95% de confianza).

> \*\*Características clave\*\*: Vesículas flotatorias presentes."

---

### \*\*6. Gestión de Cambios Típicos\*\*

Supongamos que en \*\*Sprint 4\*\* un biólogo reporta:

> "\*Las algas en aguas contaminadas pierden vesículas, confunden al sistema.\*"

\*\*Respuesta integrada\*\*:

1. \*\*CommonKADS\*\*:

- Añadir excepción: `SI (agua == "contaminada") ENTONCES descontar vesículas en Fucus`.

2. \*\*Scrum\*\*:

- Priorizar la tarea en el próximo sprint.

- Técnica: Añadir campo `calidad\_agua` a la interfaz.

---

### \*\*7. Beneficios Clave de la Combinación\*\*

- \*\*Calidad científica\*\*: Cada regla es validada por biólogos \*en el sprint\*.

- \*\*Flexibilidad\*\*: Si se descubre una nueva especie (ej: \*Alga invasora X\*), se añade como historia de usuario.

- \*\*Eficiencia\*\*:

- Ingenieros de conocimiento trabajan 1 sprint adelantados a los desarrolladores.

- La ontología de algas se reutiliza en apps móviles o APIs futuras.

---

### \*\*Conclusión\*\*

Para un SE de identificación de algas:

- \*\*CommonKADS\*\* estructura el conocimiento taxonómico complejo en reglas/ontologías validadas.

- \*\*Scrum\*\* gestiona la implementación incremental, priorizando especies por relevancia ecológica o frecuencia.

\*\*Recomendación final\*\*:

> "Comienza con algas de alta prevalencia (ej: 10 especies) y agrega atributos gradualmente. Usa \*\*fotos georreferenciadas\*\* en la validación para cubrir variaciones geográficas. ¡Involucra a biólogos en cada sprint review!"

¿Necesitas el diagrama de la ontología completa o un ejemplo del backlog en Jira? 😊

Un ejemplo del backlog en Jira

### Ejemplo de Backlog en Jira para Sistema Experto de Identificación de Algas (CommonKADS + Scrum)

---

#### \*\*Configuración del Proyecto en Jira\*\*

- \*\*Nombre del Proyecto\*\*: AlgaExpert - Sistema de Identificación de Algas Marinas

- \*\*Workflow Personalizado\*\*:

```

To Do → In Progress → Knowledge Validation → Technical Implementation → Testing → Done

```

- \*\*Campos Personalizados\*\*:

- `Tipo de Tarea`: [Conocimiento | Técnica | Híbrida]

- `Validado por Biólogo`: [Sí | No | Pendiente]

- `Precisión Estimada`: (campo numérico)

- `Especie Asociada`: (lista desplegable)

---

### \*\*Backlog Priorizado (Ejemplo para Sprint 1-2)\*\*

#### \*\*Épica: Identificación de Algas Pardas (Phaeophyceae)\*\*

\*Descripción: Implementar 5 especies comunes de algas pardas en costas ibéricas\*

---

#### \*\*Historias de Usuario (User Stories)\*\*

| Key | Tipo | Título | Descripción | Criterios de Aceptación | Tipo Tarea | Especie Asociada |

|------|------------|--------|-------------|-------------------------|------------|------------------|

| \*\*KM-1\*\* | Story | Como biólogo, quiero modelar reglas para \*Fucus vesiculosus\* | Extraer características clave: vesículas flotatorias, color pardo, hábitat intermareal | - 10 reglas validadas<br>- Precisión >90% en 30 muestras | Conocimiento | Fucus vesiculosus |

| \*\*TEC-1\*\* | Story | Como usuario, quiero ingresar características morfológicas | Formulario con campos: color, forma, textura, hábitat | - Formulario responsive<br>- Validación en tiempo real | Técnica | - |

| \*\*KM-2\*\* | Task | Adquirir conocimiento de \*Laminaria digitata\* | Entrevista con experto Dr. Marina (2 horas) | - Documentar 5 reglas clave<br>- Registrar excepciones | Conocimiento | Laminaria digitata |

| \*\*TEC-2\*\* | Task | Implementar motor de reglas básico | Usar Drools + Spring Boot | - Ejecutar reglas desde BD<br>- Tiempo respuesta <1s | Técnica | - |

---

#### \*\*Tareas de Conocimiento (CommonKADS)\*\*

| Key | Título | Descripción | Validado por Biólogo | Precisión Estimada |

|------|--------|-------------|----------------------|-------------------|

| KM-1.1 | Definir atributos para \*Fucus vesiculosus\* | Color, forma vesículas, textura, hábitat | Sí | - |

| KM-1.2 | Crear regla: identificación por vesículas | "SI tiene\_vesículas=VERDADERO Y color='pardo' ENTONCES especie=Fucus" | Sí | 92% |

| KM-1.3 | Validar con muestras del herbario | Probar con 30 especímenes digitalizados | Pendiente | 85% |

| KM-2.1 | Documentar variación estacional | Cambios morfológicos en invierno/verano | No | - |

---

#### \*\*Tareas Técnicas (Scrum)\*\*

| Key | Título | Descripción | Enlace KM |

|------|--------|-------------|-----------|

| TEC-1.1 | Diseñar UI formulario | Wireframes para ingreso de características | KM-1.1 |

| TEC-1.2 | Implementar campo "Hábitat" | Selector: [Rocas intermareales, Fondos arenosos, ...] | KM-1.1 |

| TEC-2.1 | Configurar Drools | Conexión con base de datos PostgreSQL | KM-1.2 |

| TEC-2.2 | Crear API para ejecutar reglas | Endpoint POST /api/identify | KM-1.2 |

---

### \*\*Tablero Scrum (Sprint 1)\*\*

\*\*Objetivo del Sprint\*\*: Identificar \*Fucus vesiculosus\* con 85% de precisión

| Tarea | Estado | Asignado | Validado |

|-------|--------|----------|----------|

| KM-1.1 | ✅ Done | Ing. Conocimiento | Sí |

| TEC-1.1 | ✅ Done | Frontend Dev | - |

| KM-1.2 | ⏳ In Progress | Ing. Conocimiento | Sí |

| TEC-2.1 | ⏳ In Progress | Backend Dev | - |

| KM-1.3 | ◻️ To Do | Biólogo | Pendiente |

---

### \*\*Definición de Done (DoD)\*\*

Para tareas de \*\*conocimiento\*\*:

✅ Regla documentada en formato estándar

✅ Validada por al menos un biólogo

✅ Precisión >85% en pruebas

Para tareas \*\*técnicas\*\*:

✅ Código en repositorio Git

✅ Pruebas unitarias aprobadas

✅ Integrado en ambiente de staging

---

### \*\*Sprint Review (Ejemplo)\*\*

\*\*Demostración\*\*:

1. Mostrar formulario de ingreso (TEC-1.1)

2. Ejecutar regla KM-1.2 con datos de prueba:

```json

{"color": "pardo", "tiene\_vesículas": true, "habitat": "roca"}

```

3. Resultado en UI:

> "\*\*Especie probable\*\*: Fucus vesiculosus (92% confianza)"

\*\*Validación biólogo\*\*:

> "La regla KM-1.2 identificó correctamente 28/30 muestras. Ajustar para algas juveniles sin vesículas"

\*\*Nueva tarea backlog\*\*:

```markdown

KM-1.4 - Añadir excepción para algas juveniles:

"SI longitud < 15cm ENTONCES solicitar análisis adicional"

```

---

### \*\*Plantilla de Issue en Jira\*\*

```markdown

\*\*Epic\*\*: [Identificación de Algas Pardas]

\*\*Tipo\*\*: [Story/Task]

\*\*Tipo Tarea\*\*: [Conocimiento/Técnica/Híbrida]

\*\*Descripción\*\*:

[Detalle de la tarea]

\*\*Criterios Aceptación\*\*:

- [ ] Requisito 1

- [ ] Requisito 2

\*\*Validación Biológica\*\*:

- [ ] Sí

- [ ] No

- [ ] Pendiente

\*\*Especies Asociadas\*\*: [Fucus vesiculosus]

```

---

### \*\*Ventajas de Esta Estructura\*\*

1. \*\*Trazabilidad completa\*\*: Cada implementación técnica (TEC-x) vincula a su regla de conocimiento (KM-x)

2. \*\*Validación integrada\*\*: Campo "Validado por Biólogo" en cada tarea KM

3. \*\*Priorización visual\*\*: Especies asociadas permiten filtrar por importancia ecológica

4. \*\*Métrica clave\*\*: Precisión estimada como KPI de calidad de conocimiento

\*\*Recomendación\*\*: Usar \*Jira Automation\* para:

- Notificar a biólogos cuando tareas KM necesitan validación

- Bloquear tareas TEC si su KM asociado no está validado

¿Necesitas una exportación CSV de este backlog para importar en tu Jira? 😊